# Comparing Structure Learning Algorithms Across Different Libraries

**Tian Cheng Xia**
Master's Degree in Artificial Intelligence, University of Bologna
tiancheng.xia@studio.unibo.it

January 28, 2024

**Abstract**

This mini-project aims to explore and test different structure learning algorithms available in the existing open-source libraries for Bayesian networks. We reviewed the available Python libraries and experimented with the algorithms available in four of them, evaluating the methods on classification tasks. We empirically observed that networks learned using tree-based methods are less accurate but inherently simpler while constraint-based and score-based methods are more accurate but result in more complex topologies.

## Introduction

### Domain

Structure learning algorithms are methods to infer the topology of a Bayesian network through independence tests or fitness scores. Various algorithms have been proposed in the literature and in this mini-project we explore what has been implemented in the existing Python libraries.

### Aim

In this mini-project, we aim to review existing open-source Python libraries that implement structure learning algorithms and compare them on classification problems.

### Method

We evaluate the available structure learning algorithms on the task of classification. For training and testing, we use the following datasets where continuous features have been discretized:

- *Apple quality*, with 7 Gaussian distributed features.
- *Heart diseases*, with 13 features, mostly categorical.

The libraries we experimented with are listed in Table 1. Other libraries (`PyOpenPNL`, `pyBN`, `libpgm`) have been considered but have been discarded as they are outdated and not actively maintained.

### Results

We empirically observed that among the various classes of structure learning algorithms, tree-based methods are less accurate but produce a smaller network while constraint-based and score-based methods achieve higher accuracy at the cost of a more complex model.

| Library | Version | Backend | License |
|---|---|---|---|
| `pgmpy` | 0.1.24 | Numpy | MIT |
| `bnlearn` | 0.8.4 | `bnlearn` | MIT |
| `pomegranate` | 1.0.3 | PyTorch | MIT |
| `pyAgrum` | 1.11.0 | C++ | GPLv3 |

Table 1: Overview of the experimented libraries.

## Model

As we are interested in testing structure learning algorithms, we infer the topology of the networks using these methods and, for a more straightforward and fair comparison, we always use MLE as parameter learning algorithm and make queries only using exact inference through variable elimination.

The algorithms available in each library we experimented with are listed in Table 2. When allowed, we tested different fitness functions and independence tests.

## Analysis

### Experimental setup

Each dataset has been split into a train and test set. The train set is fed to the structure learning algorithms to determine the topology of the network. The test set is used to evaluate the resulting models by querying the target variable given as evidence the other features.

For each structure learning algorithm, we keep track of the time required to learn the structure and the total number of edges in the computed model.

### Results

Results on the *apple quality* dataset are shown in Figure 1. The highest accuracy has been achieved by two score-based algorithms: hill-climbing and tabu search, while tree-based methods are those that obtained the lowest accuracy. On the other hand, analyzing the complexity of the learned network, intended as the number of edges, we can observe that tree-based methods, as expected, produced simpler topologies, while A* and hill-climbing resulted in bigger models.

Results[1] on the *heart diseases* dataset are presented in

[1] A* and MMHC are omitted as computationally too expensive.

| Library | Structure learning algorithms | | | |
|---|---|---|---|---|
| | Score-based | Tree-based | Constraint-based | Hybrid |
| `pgmpy` `bnlearn` | Hill-climbing<br>Exhaustive search | Chow-Liu<br>Naive Bayes<br>Tree-augmented NB | PC | Max-min hill-climbing |
| `pomegranate` | A*[a] | Chow-Liu[a] | — | — |
| `pyAgrum` | Hill-climbing<br>Tabu search<br>K2 | Chow-Liu[b]<br>Naive Bayes[b]<br>Tree-augmented NB[b] | MIIC | — |

[a]Not explicitly listed in the official documentation.
[b]Only available for Bayesian network classifiers.

Table 2: Structure learning algorithms available in each library.

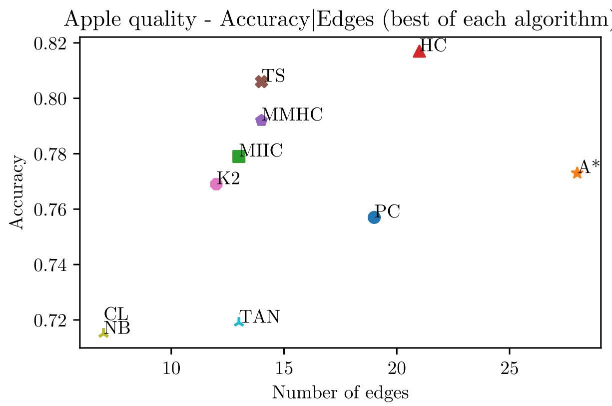


Figure 1: Relationship between accuracy and number of edges on the *apple quality* dataset.

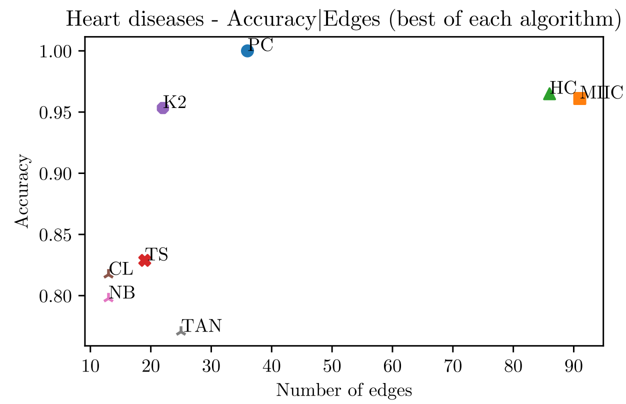


Figure 2: Relationship between accuracy and number of edges on the *heart diseases* dataset.

Figure 2. In this case, the PC algorithm reached the highest accuracy, while, again, tree-based methods have the lowest performance. Hill-climbing is still one of the best-performing methods but it is also among the algorithms that resulted in the most complex topology.

Overall[2], we can empirically observe that hill-climbing is the most consistent algorithm across the two datasets but lacks the capability of creating a simpler topology. Tree-based methods are instead those with the lowest, but still acceptable, results and have the advantage of learning smaller and more interpretable networks.

Compared to the literature, our results are in line with what is observed in (Constantinou et al. 2021). In fact, the performances of the various structure learning algorithms depend on the data and it is not possible to designate one as the best one.

## Conclusion

In this mini-project, we explored the structure learning algorithms available in four Python libraries. We evaluated these algorithms on two classification datasets and empirically observed that it is not reasonable to label one of them as the best one. From a more general point of view, we concluded that tree-based methods produce less accurate but simpler networks while constraint-based and score-based methods are more accurate but result in more complex topologies.

[2]For more details regarding our experiment, we refer the reader to the `plots.ipynb` notebook.

## Links to external resources

- *Apple quality* dataset: `https://www.kaggle.com/datasets/nelgiriyewithana/apple-quality`.
- *Heart diseases* dataset: `https://www.kaggle.com/datasets/ineubytes/heart-disease-dataset`.

## References


Constantinou, A. C.; Liu, Y.; Chobtham, K.; Guo, Z.; and Kitson, N. K. 2021. Large-scale empirical validation of bayesian network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning* 131:151–188.